AD-A053 022 SYSTEMS RESEARCH LABS INC NEWPORT NEWS VA RASA DIV F/G 1/3
DOCUMENTATION OF HELICOPTER AEROELASTIC STABILITY ANALYSIS COMP--ETC(U)
DEC 77 L R SUTTON DAAJ02-76-C-0032
UNCLASSIFIED RASA/SRL-14-77-04 USARTL-TR-77-52 NL

5 OF 5
AD
A053022

END
DATE
FILMED
6 -78
DDC

```
      IF(MFEA.EQ.0) GO TO 32
      TKN(KT1)=SMLA(MS)*YKM(I,MS)*CTB2(L)
      KT1=KT1+NRIFC
   32 IF(MFLAP.EQ.0) GO TO 34
      TKN(KT1)=SMLA(MS)*CTB3(L)*YKM(I,MS)
      KT1=KT1+NRIFC
   34 IF(MLEL.EQ.0) GO TO 36
      TKN(KT1)=SMLA(MS)*CTB4(L)*YKM(I,MS)
   36 KT=KT+1
   40 IF(MFEA.EQ.0) GO TO 50
      KT2=KT+NRIFC
      IF(MCT.EQ.0) GO TO 42
      TKN(KT)=-FAB1(L)
      KT2=KT2+NRIFC
   42 IF(MFLAP.EQ.0) GO TO 44
      TKN(KT2)=-FAB3(L)
      KT2=KT2+NRIFC
   44 IF(MLEL.EQ.0) GO TO 46
      TKN(KT2)=-FAB4(L)
   46 KT=KT+1
   50 IF(MFLAP.EQ.0) GO TO 56
      KT1=KT
      IF(MCT.EQ.0) GO TO 52
      TKN(KT)=-FLB1(L)
      KT1=KT1+NRIFC
   52 IF(MFEA.EQ.0) GO TO 54
      TKN(KT1)=-FLB2(L)
      KT1=KT1+NRIFC
   54 KT1=KT1+NRIFC
      IF(MLEL.EQ.0) GO TO 55
      TKN(KT1)=-FLB4(L)
   55 KT=KT+1
   56 IF(MLEL.EQ.0) GO TO 60
      IF(MCT.EQ.0) GO TO 57
      TKN(KT)=-FSB1(L)
      KT=KT+NRIFC
   57 IF(MFEA.EQ.0) GO TO 58
      TKN(KT)=-FSB2(L)
      KT=KT+NRIFC
   58 IF(MFLAP.EQ.0) GO TO 60
      TKN(KT)=-FSB3(L)
   60 CONTINUE
  900 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SUBMF(I,J)
      INTEGER P,Q
      COMPLEX EXPON
      COMPLEX*16 TKN(441)
      COMMON/INTER/NSY,NBSEC,NFSEC,NB,NBP,MFLAP,MFEA,MCT,
     1MFLEX,MCON,MAER,MFUS,NBC,NFLAP,NFEA,NCT,NCON,NFFB,
     2NAS,NHC,NVI,NSP,MAXN,NES,MSC,NEGN,IPCT,NIT,MER,NORM,
     3IREM,NEX,NPS,NSCH,IG,IF,NPRL,NPRS,NPD,NSK,NCOLS,NCSB,
     4NFP1,MXSMI,MXT2P1,MXKQ,MXCPL,MXCSB,MXCPM,MXCPK,MXSMB,
     5NEBC,NESBC,MFASB,MXFAB,NFUS,NRBD,NRIFC,MXQ,NEIFC,
     6NEISC,NEITC,MXTKN,NFF,MINPN,MAXPN,IBF,MODE
      COMMON/TKN1/TKN
      COMMON/NLEAD/MLEL,NLEL,MCTY
      IF(NB.EQ.1)GO TO 586
      NMK=J-I
      DO 547 MS=2,NB
      DO 547 Q=1,NRBD
      DO 547 P=1,NRBD
      KT=NEIFC+NEISC+(Q-1)*NRIFC+NFUS+MXT2P1+P
      KT2=KT+(MS-1)*(NEITC+NRBD)
  547 TKN(KT2)=TKN(KT)*EXPON(NMK,MS)
  586 RETURN
      END
```

```
      SUBROUTINE SOLVE
      REAL*8 DTLG10,DFAC,FAC
      COMPLEX*16 ZERO,TWO,SUM,SWAP,DTPHAS,DPIVOT,DETSV
      COMPLEX*16 EPS(63),FORCE(63),DB(63,63)
      COMMON/EPSA/EPS
      COMMON/REMA/DETSV
      COMMON/CDETRM/DPIVOT,DTPHAS,DTLG10,IDET
      COMMON/DTERM/DFAC
      FAC=20.D0
      TWO=DCMPLX(2.D0,0.D0)
      ZERO=DCMPLX(0.D0,0.D0)
      REWIND 1
      READ(1)MXSMI,NRIFC,NHC,NORM,IREM1,NEX
      NORDER=MXSMI*NRIFC
      DO 11 L=1,MXSMI
      IROWF=L*NRIFC
      IROWS=IROWF-NRIFC+1
      DO 11 K=1,MXSMI
      ICOLF=K*NRIFC
      ICOLS=ICOLF-NRIFC+1
   11 READ(1)((DB(I,J),I=IROWS,IROWF),J=ICOLS,ICOLF)
      REWIND 1
      DO 12 I=1,NORDER
      IK=NORDER+1-I
   12 FORCE(IK)=EPS(IK)
C     DO 200 J=1,45
C 200 WRITE(6,300)(DB(I,J),I=1,45)
C 300 FORMAT(5X,10D12.4)
      IDET=1
      ISBLK=NHC*NRIFC
      ICOL=ISBLK+NORM
      IROW=ISBLK+IREM1
      NEXCOL=ISBLK+NEX
      CALL SWAPS(DB,NORDER,ICOL,IROW)
      SWAP=FORCE(IROW)
      FORCE(IROW)=FORCE(NORDER)
      FORCE(NORDER)=SWAP
      N=NORDER-1
      CALL ERRSET(208,500)
      CALL DCMAT(DB,N,FORCE)
      CALL ERRSET(208,10)
      WRITE(6,102)DTPHAS,DTLG10,DPIVOT
  102 FORMAT(//2X,'DTPHAS = ',2D40.16,/2X,'DTLG10 = ',
     1D40.16,/2X,'DPIVOT = ',2D40.16)
      IFAC=DTLG10/FAC
      DFAC=IFAC*FAC
      DTLG10=DTLG10-DFAC
      DTPHAS=DTPHAS*DCMPLX(10.D0**DTLG10,0.D0)
      DETSV=DTPHAS*DPIVOT
      SUM=ZERO
```

```
      DO 2 I=1,N
      IF(I.EQ.NEXCOL) GO TO 2
      SUM=SUM+DB(NORDER,I)*FORCE(I)
    2 CONTINUE
      SUM=FORCE(NORDER)-SUM
      IF(CDABS(DB(NORDER,NEXCOL)).NE.0.D0)GO TO 5
      WRITE(6,6)NORDER,NEXCOL
    6 FORMAT('0    DB(',I5,' ,',I5,' )    IS ZERO')
      STOP
    5 CONTINUE
      SUM=SUM/DB(NORDER,NEXCOL)
      FORCE(NEXCOL)=(SUM+FORCE(NEXCOL))/TWO
      FORCE(NORDER)=FORCE(ICOL)
      FORCE(ICOL)=ZERO
      DO 3 I=1,NORDER
    3 EPS(I)=FORCE(I)
      RETURN
      END
```

```
      SUBROUTINE SWAPS(DB,NORDER,ICOL,IROW)
      COMPLEX*16 DB(63,63),SWAP
      DO 1 I=1,NORDER
      SWAP=DB(I,ICOL)
      DB(I,ICOL)=DB(I,NORDER)
    1 DB(I,NORDER)=SWAP
      DO 2 I=1,NORDER
      SWAP=DB(IROW,I)
      DB(IROW,I)=DB(NORDER,I)
    2 DB(NORDER,I)=SWAP
      RETURN
      END
```

```
      SUBROUTINE DCMAT(A,N,Y)
      REAL*8 ADET,AMAG,DSIGN
      COMPLEX*16 A,Y,DDET,DPIVOT,TK,X
      COMPLEX*16 DAIJ,AMX,DONE,DYI,TEMP,DAKJ,DYK,DAKK,DAIK
     1,ONE,DPHAS
      DIMENSION ICHG(63),A(63,63),Y(63),X(63)
      COMMON /CDETRM/ DPIVOT,DPHAS,ADET,IDET
      NDIM=63
      ADET=0.D0
      DSIGN=1.D0
      DPHAS=DCMPLX(1.0D0,0.D0)
      NP1=N
      IF(IDET.EQ.0) GO TO 650
      NP1=N+1
      DO 651 I=1,NP1
  651 X(I)=A(NP1,I)
  650 CONTINUE
      DO 118 K=1,N
      AMX = A(K,K)
      IMX=K
      DO 100 I=K,N
      IF(CDABS(A(I,K)) .LE. CDABS(AMX)) GO TO 100
      AMX = A(I,K)
      IMX=I
  100 CONTINUE
  102 IF (IMX.EQ.K) GO TO 106
      DO 104 J=1,NP1
      TEMP=A(K,J)
      A(K,J)=A(IMX,J)
  104 A(IMX,J)=TEMP
      ICHG(K)=IMX
      TEMP=Y(K)
      Y(K)= Y(IMX)
      Y(IMX)= TEMP
      DPHAS=-DPHAS
      GO TO 108
  106 ICHG(K)=K
  108 CONTINUE
      DAKK=A(K,K)
  901 FORMAT(1X,I5,2D40.16/)
C     WRITE(6,1000) DAKK
C1000 FORMAT(5X,'DAKK',5X,2D20.10)
      AMAG=CDABS(DAKK)
      IF(AMAG.NE.0.D0) GO TO 6
      WRITE(6,7)
    7 FORMAT('0   MATRIX IN DCMAT IS SINGULAR')
      STOP
    6 CONTINUE
      ADET=ADET+DLOG10(AMAG)
      DPHAS=DPHAS*DAKK/AMAG
```

```
      DONE=DCMPLX(1.D0,0.D0)
      DAKK=DONE/DAKK
      DO 110 J=1,NP1
  110 A(K,J)=A(K,J)*DAKK
      A(K,K)=DAKK
      IF(IDET.EQ.0) GO TO 652
      TK=X(K)
      DO 653 J=K,NP1
  653 X(J)=X(J)-TK*A(K,J)
  652 CONTINUE
      DYK=Y(K)
      Y(K)=DYK*DAKK
      DO 114 I=1,N
      IF (I.EQ.K) GO TO 114
      DAIK=A(I,K)
      CALL ROWSUM(NP1,NDIM,A(I,1),A(K,1),DAIK)
C     DO 112 J=1,NP1
C 112 A(I,J)=A(I,J)-DAIK*A(K,J)
      A(I,K)=DAIK
      DYI=Y(I)
      DYK=Y(K)
      Y(I)=DYI-DAIK*DYK
  114 CONTINUE
      DO 116 I=1,N
  116 A(I,K)=-A(I,K)*DAKK
      A(K,K)=DAKK
  118 CONTINUE
      DO 122 K=1,N
      L=N+1-K
      KI=ICHG(L)
      IF (L.EQ.KI) GO TO 122
      DO 120 I=1,N
      TEMP = A(I,L)
      A(I,L) = A(I,KI)
  120 A(I,KI) = TEMP
  122 CONTINUE
      IF(IDET.NE.0) DPIVOT=X(NP1)
  124 RETURN
      END
```

```
      COMPLEX FUNCTION EXPON(L,MS)
C  CREATE EXP(I*L*PHIM)
      DIMENSION CS(4,6),SN(4,6)
      COMMON/RNAM/CS,SN
      IL=IABS(L)
      IF(L) 16,15,17
   15 EXPON=CMPLX(1.,0.)
      GO TO 18
   16 A=CS(MS,IL)
      B=SN(MS,IL)
      EXPON=CMPLX(A,-B)
      GO TO 18
   17 A=CS(MS,L)
      B=SN(MS,L)
      EXPON=CMPLX(A,B)
   18 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SWA(I,J)
      INTEGER P,Q,QS
      COMPLEX EXPON
      COMPLEX*16 CMS,CM1,CS1,CS2
      COMPLEX*16 TKN(441)
      COMPLEX*16 ZLN
      COMPLEX*16 XNLQ
      COMPLEX*16 YKM(3,4)
      COMMON/SPAR/AKCI(6),TAU(6),SMLA(6),DMS(6),AK(4),AC(4),BJ(4),
     1CAPK,CAPC
      COMMON/INTER/NSY,NBSEC,NFSEC,NB,NBP,MFLAP,MFEA,MCT,
     1MFLEX,MCON,MAER,MFUS,NBC,NFLAP,NFEA,NCT,NCON,NFFB,
     2NAS,NHC,NVI,NSP,MAXN,NES,MSC,NEGN,IPCT,NIT,MER,NORM,
     3IREM,NEX,NPS,NSCH,IG,IF,NPRL,NPRS,NPD,NSK,NCOLS,NCSB,
     4NFP1,MXSMI,MXT2P1,MXKQ,MXCPL,MXCSB,MXCPM,MXCPK,MXSMB,
     5NEBC,NESBC,MFASB,MXFAB,NFUS,NRBD,NRIFC,MXQ,NEIFC,
     6NEISC,NEITC,MXTKN,NFF,MINPN,MAXPN,IBF,MODE
      COMMON/NLEAD/MLEL,NLEL,MCTY
      COMMON/ROTF/OM1,OM2,OMT
      COMMON/FREF/CMS,CM1,CS1,CS2
      COMMON/SWASH/SWGJ,SWEI,SWM,SWR
      COMMON/TKN1/TKN
      KSML=I-NFP1
      IF(I.NE.J) GO TO 30
      DO 17 L=1,MXT2P1
      LS=L-MAXN-1
      LL=(L-1)*NRIFC+L
   17 TKN(LL)=ZLN(LS,I)
      DO 20 MS=1,NB
      DO 20 L=1,MXT2P1
      LS=L-MAXN-1
      CFDL=1.0+DMS(MS)*(1.+(LS*LS-1)/(1.+LS*LS*SWGJ/SWEI))/SWR
      LL=(L-1)*NRIFC+MXT2P1+(MS-1)*NRBD+NCOLS+NFUS
      IF(MFLEX.EQ.0) GO TO 21
      IF(MCON.EQ.0) GO TO 20
      LL=LL+3
      IF(MCTY.GT.0) LL=LL+3
      TKN(LL)=-EXPON(LS,MS)*CFDL
      GO TO 20
   21 LL=LL+MCT
      IF(MCT.EQ.0) GO TO 20
      CS1=CMS-CM1*KSML*OM1
      YKM(I,MS)=SMLA(MS)*(1+CS1*TAU(MS))
      TKN(LL)=YKM(I,MS)*EXPON(LS,MS)*CFDL
   20 CONTINUE
      GO TO 50
   30 IMJ=I-J
      DO 18 L=1,MXT2P1
      DO 18 Q=1,MXT2P1
      IF(L.EQ.Q) GO TO 18
```

```
      LMQ=L-Q
      IF(IMJ.NE.LMQ) GO TO 18
      LS=L-MAXN-1
      QS=Q-MAXN-1
      LL=(L-1)*NRIFC+Q
      TKN(LL)=XNLQ(I,LS,QS)
   18 CONTINUE
   50 RETURN
      END
```

```
      SUBROUTINE SWB(I,J)
      INTEGER P,Q,QS
      REAL*6 CX(75)
      COMPLEX EXPON
      COMPLEX*16 ULN,S(216)
      COMPLEX*16 TKN(441)
      COMPLEX*16 B(648),SMLB(108),SMLC(108),SMLD(108)
      COMPLEX*16 CTB(54),FAB(54),FLB(54)
      COMPLEX*16 CTB1(9),CTB2(9),CTB3(9),FAB1(9),FAB3(9),FLB1(9),FLB2(9)
      COMPLEX*16 SMLE(108),FSB(54)
      COMPLEX*16 SMLF(108),SMLG(108)
      COMPLEX*16 FSB1(9),FSB2(9),FSB3(9),FAB4(9),FLB4(9),CTB4(9)
      COMMON/BTS/B,SMLB,SMLC,SMLD,CTB,FAB,FLB,CTB1,CTB2,CTB3,FAB1,
     1FAB3,FLB1,FLB2,SMLE,FSB,FSB1,FSB2,FSB3,FAB4,FLB4,CTB4,SMLF,SMLG
      COMMON/SWASH/SWGJ,SWEI,SWM,SWR
      COMMON/INTER/NSY,NBSEC,NFSEC,NB,NBP,MFLAP,MFEA,MCT,
     1MFLEX,MCON,MAER,MFUS,NBC,NFLAP,NFEA,NCT,NCON,NFFB,
     2NAS,NHC,NVI,NSP,MAXN,NES,MSC,NEGN,IPCT,NIT,MER,NORM,
     3IREM,NEX,NPS,NSCH,IG,IF,NPRL,NPRS,NPD,NSK,NCOLS,NCSB,
     4NFP1,MXSMI,MXT2P1,MXKQ,MXCPL,MXCSB,MXCPM,MXCPK,MXSMB,
     5NEBC,NESBC,MFASB,MXFAB,NFUS,NRBD,NRIFC,MXQ,NEIFC,
     6NEISC,NEITC,MXTKN,NFF,MINPN,MAXPN,IBF,MODE
      COMMON/SPAR/AKCI(6),TAU(6),SMLA(6),DMS(6),AK(4),AC(4),BJ(4),
     1CAPK,CAPC
      COMMON/CFLEX/CX
      COMMON/TKN1/TKN
      COMMON/NLEAD/MLEL,NLEL,MCTY
      COMMON/SS1/S
      NONO=1
      IMJ=I-J
      JMNC=(J-1)*12*NCOLS
      IF(MFLEX.EQ.0.AND.IMJ.NE.0) GO TO 23
      IF(MFLEX.EQ.0) GO TO 3
      IF(MCTY.EQ.0) GO TO 3
      IF(IMJ.NE.0) GO TO 23
    3 IF(IMJ.NE.0) NONO=0
      IF(NBP.EQ.0) GO TO 13
      KSML=I-NFP1
      NPMK=-NPS-KSML
      DO 14 Q=1,MXT2P1
      QS=Q-MAXN-1
      NMKQ=NPMK-QS
      IF(NMKQ.EQ.0) GO TO 5
      RNMKQ=1.0*NMKQ
      RFA=RNMKQ/NBP
      NFA=NMKQ/NBP
      DIF=RFA-1.0*NFA
      IF(DIF.GE.0.0) GO TO 2
      DIF=-DIF
    2 IF(DIF.GT..05) GO TO 14
```

```
  5 DFQR=1.0+DMS(1)*(1.+(QS*QS-1.)/(1.+QS*QS*SWGJ/SWEI))/SWR
    IF(MFLEX.EQ.0) GO TO 7
    IF(MCON.EQ.0) GO TO 14
    IF(MCTY.GT.0) GO TO 6
    DO 8 IQ=1,NCOLS
    LL=NEIFC+NEISC+(IQ-1)*NRIFC+Q
    L=(J-1)*NCOLS*MXSMI+(I-1)*NCOLS+IQ
  8 TKN(LL)=-NBP*DFQR*FLB(L)
    LL=NEIFC+NEISC+NCOLS*NRIFC+Q
    TKN(LL)=+NBP*DFQR*CX(3)*NONO
    TKN(LL+NRIFC)=+NBP*DFQR*CX(8)*NONO
    TKN(LL+NRIFC*2)=-NBP*DFQR*CX(12)*NONO
    GO TO 14
  6 LL=NEIFC+NEISC+NRIFC*NCOLS+5*NRIFC+Q
    TKN(LL)=-NBP*DFQR
    GO TO 14
  7 LL=NEIFC+NEISC+NRIFC*NCOLS+Q
    IF(MCT.EQ.0) GO TO 14
    TKN(LL)=-NBP*DFQR/SMLA(1)
 14 CONTINUE
    GO TO 23
 13 DO 21 MS=1,NB
    DO 21 P=1,MXT2P1
    QS=P-MAXN-1
    DFQR=1.+DMS(MS)*(1+(QS*QS-1)/(1.+QS*QS*SWGJ/SWEI))/SWR
    IF(MFLEX.EQ.0) GO TO 22
    IF(MCON.EQ.0) GO TO 21
    IF(MCTY.GT.0) GO TO 25
    DO 24 IQ=1,NCOLS
    LL=NEIFC+NEISC+(MS-1)*NEITC+(IQ-1)*NRIFC+P
    L=(J-1)*NCOLS*MXSMI+(I-1)*NCOLS+IQ
 24 TKN(LL)=-EXPON(-QS,MS)*DFQR*FLB(L)
    LL=NEIFC+NEISC+(MS-1)*NEITC+NCOLS*NRIFC+P
    TKN(LL)=+EXPON(-QS,MS)*DFQR*CX(3)*NONO
    LL=LL+NRIFC
    TKN(LL)=+EXPON(-QS,MS)*DFQR*CX(8)*NONO
    LL=LL+NRIFC
    TKN(LL)=-EXPON(-QS,MS)*DFQR*CX(12)*NONO
    GO TO 21
 25 LL=NEIFC+NEISC+(MS-1)*NEITC+NRIFC*NCOLS+5*NRIFC+P
    TKN(LL)=-EXPON(-QS,MS)*DFQR
    GO TO 21
 22 LL=NEIFC+NEISC+(MS-1)*NEITC+NRIFC*NCOLS+P
    IF(MCT.EQ.0) GO TO 21
    TKN(LL)=-EXPON(-QS,MS)*DFQR/SMLA(MS)
 21 CONTINUE
 23 CONTINUE
    IF(MFUS.EQ.0) GO TO 52
    JMI=-IMJ
    DO 50 IPP=1,MXT2P1
```

```
      LS=IPP-MAXN+1
      IF(JMI.NE.LS) GO TO 50
      DO 51 IQQ=1,NCOLS
      LLP=JMNC+(IQQ-1)*12+1
      LL=NEIFC+(IQQ-1)*NRIFC+IPP
   51 TKN(LL)=ULN(LS,KSML)*S(LLP)
   50 CONTINUE
   52 CONTINUE
      RETURN
      END
```

```
      COMPLEX FUNCTION ZLN*16(LS,I)
      REAL*8 C3,C4
      COMPLEX EXCHI
      COMPLEX*16 CLNJ,C1,C2,C5,C6,C7,C8,C9,C10
      COMPLEX*16 CMS,CM1,CS1,CS2,VN,VN1,VN2
      COMMON/SWASH/SWGJ,SWEI,SWM,SWR
      COMMON/INTER/NSY,NBSEC,NFSEC,NB,NBP,MFLAP,MFEA,MCT,
     1MFLEX,MCON,MAER,MFUS,NBC,NFLAP,NFEA,NCT,NCON,NFFB,
     2NAS,NHC,NVI,NSP,MAXN,NES,MSC,NEGN,IPCT,NIT,MER,NORM,
     3IREM,NEX,NPS,NSCH,IG,IF,NPRL,NPRS,NPD,NSK,NCOLS,NCSB,
     4NFP1,MXSMI,MXT2P1,MXKQ,MXCPL,MXCSB,MXCPM,MXCPK,MXSMB,
     5NEBC,NESBC,MFASB,MXFAB,NFUS,NRBD,NRIFC,MXQ,NEIFC,
     6NEISC,NEITC,MXTKN,NFF,MINPN,MAXPN,IBF,MODE
      COMMON/SPAR/AKCI(6),TAU(6),SMLA(6),DMS(6),AK(4),AC(4),BJ(4),
     1CAPK,CAPC
      COMMON/NLEAD/MLEL,NLEL,MCTY
      COMMON/FREF/CMS,CM1,CS1,CS2
      COMMON/ROTF/OM1,OM2,OMT
      COMMON/SPAR1/AKT(4),ACT(4),AKP(4),ACP(4)
      R=SWR
      KSML=I-NFP1
      CS1=CMS-CM1*KSML*OM1
      CS2=CS1*CS1
      C1=SWM*(CS2-OMT*LS*CM1*CS1-LS*LS*OM2)
      C2=DCMPLX(0.D0,0.D0)
      C9=DCMPLX(0.D0,0.D0)
      CFL=1.+(LS*LS-1)/(1+LS*LS*SWGJ/SWEI)
      CFLR=CFL/SWR
      DO 10 JJ=1,NES
      CLNJ=AK(JJ)+CS1*AC(JJ)-CM1*LS*OM1*AC(JJ)
      C2=C2+CLNJ
   10 C9=C9+CLNJ*(1-BJ(JJ)*CFLR)
      C10=C2
      C2=C9
      IF(MSC.EQ.0) GO TO 11
      C2=C10
      C5=CAPK+CAPC*(CS1-CM1*LS*OM1)
      C6=DCMPLX(0.D0,0.D0)
      C7=DCMPLX(0.D0,0.D0)
      DO 8 JJ=1,NES
      C8=AK(JJ)+CS1* AC(JJ)-CM1*LS*OM1*AC(JJ)
      C6=C6+C8*EXCHI(LS,0,JJ)*(1.-BJ(JJ)*CFLR)
    8 C7=C7+C8*EXCHI(0,LS,JJ)*(1-BJ(JJ)*CFLR)
      C2=C9-C7*C6/(C5+C2)
   11 IF(MAXN.EQ.1) GO TO 12
      KX=1-LS*LS
      C3=2.D0*3.141592654*LS*LS*KX*KX
      C4=R*R*R*(1./SWGJ+LS*LS/SWEI)
      C3=C3/C4
      ZLN=C1+C2+DCMPLX(C3,0.D0)
```

```
      GO TO 13
   12 ZLN=C1+C2
   13 CONTINUE
      VN=DCMPLX(0.D0,0.D0)
      VN1=DCMPLX(0.D0,0.D0)
      VN2=DCMPLX(0.D0,0.D0)
      DO 5 JJ=1,NES
      VN=VN+AKT(JJ)+(CS1-CM1*LS*OM1)*ACT(JJ)
      VN1=VN1+(AK(JJ)+(CS1-CM1*LS*OM1)*AC(JJ))*BJ(JJ)*(R-BJ(JJ)*CFL)
    5 VN2=VN2+AKP(JJ)+(CS1-CM1*LS*OM1)*ACP(JJ)
      ZLN=ZLN+(LS*LS*VN-VN1*CFL+CFL*CFL*VN2)/(R*R)
      RETURN
      END
```

```
      COMPLEX FUNCTION XNLQ*16(I,LS,QS)
      INTEGER QS
      COMPLEX EXCHI
      COMPLEX*16 CS1,CS2,CM1,CMS,XN,XN1,XN2,XN3
      COMPLEX*16 C5,WN,WN1,WN2
      COMMON/SWASH/SWGJ,SWEI,SWM,SWR
      COMMON/INTER/NSY,NBSEC,NFSEC,NB,NBP,MFLAP,MFEA,MCT,
     1MFLEX,MCON,MAER,MFUS,NBC,NFLAP,NFEA,NCT,NCON,NFFB,
     2NAS,NHC,NVI,NSP,MAXN,NES,MSC,NEGN,IPCT,NIT,MER,NORM,
     3IREM,NEX,NPS,NSCH,IG,IF,NPRL,NPRS,NPD,NSK,NCOLS,NCSB,
     4NFP1,MXSMI,MXT2P1,MXKQ,MXCPL,MXCSB,MXCPM,MXCPK,MXSMB,
     5NEBC,NESBC,MFASB,MXFAB,NFUS,NRBD,NRIFC,MXQ,NEIFC,
     6NEISC,NEITC,MXTKN,NFF,MINPN,MAXPN,IBF,MODE
      COMMON/SPAR/AKCI(6),TAU(6),SMLA(6),DMS(6),AK(4),AC(4),BJ(4),
     1CAPK,CAPC
      COMMON/NLEAD/MLEL,NLEL,MCTY
      COMMON/FREF/CMS,CM1,CS1,CS2
      COMMON/ROTF/OM1,OM2,OMT
      COMMON/SPAR1/AKT(4),ACT(4),AKP(4),ACP(4)
      KSML=I-NFP1
      CS1=CMS-CM1*KSML*OM1
      IF(QS.EQ.LS) GO TO 15
      XNLQ=DCMPLX(0.D0,0.D0)
      CFQ=1.+(QS*QS-1)/(1.+QS*QS*SWGJ/SWEI)
      CFL=1.+(LS*LS-1)/(1+LS*LS*SWGJ/SWEI)
      CFQR=CFQ/SWR
      CFLR=CFL/SWR
      DO 10 JJ=1,NES
   10 XNLQ=XNLQ+(AK(JJ)+(CS1-CM1*QS*OM1)*AC(JJ))*EXCHI(LS,QS,JJ)*
     1(1.-BJ(JJ)*CFQR)
      IF(MSC.EQ.0) GO TO 16
      C5=CAPK+CAPC*(CS1-CM1*QS*OM1)
      XN1=DCMPLX(0.D0,0.D0)
      XN2=DCMPLX(0.D0,0.D0)
      XN3=DCMPLX(0.D0,0.D0)
      DO 12 JJ=1,NES
      XN=AK(JJ)+(CS1-CM1*QS*OM1)*AC(JJ)
      XN1=XN1+XN
      XN2=XN2+XN*EXCHI(LS,0,JJ)*(1.-BJ(JJ)*CFLR)
   12 XN3=XN3+XN*EXCHI(0,QS,JJ)*(1.-BJ(JJ)*CFQR)
      XNLQ=XNLQ-XN3*XN2/(C5+XN1)
      GO TO 16
   15 XNLQ=DCMPLX(0.D0,0.D0)
   16 CONTINUE
      R=SWR
      WN =DCMPLX(0.D0,0.D0)
      WN1=DCMPLX(0.D0,0.D0)
      WN2=DCMPLX(0.D0,0.D0)
      DO 5 JJ=1,NES
      WN=WN+(AKT(JJ)+(CS1-CM1*QS*OM1)*ACT(JJ))*EXCHI(LS,QS,JJ)
```

```
      WN1=WN1+(AK(JJ)+(CS1-CM1*QS*OM1)*AC(JJ))*BJ(JJ)*(R-BJ(JJ)*CFQ)*
     1 EXCHI(LS,QS,JJ)
    5 WN2=WN2+(AKP(JJ)+(CS1-CM1*QS*OM1)*ACP(JJ))*EXCHI(LS,QS,JJ)
      XNLQ=XNLQ+(QS*LS*WN-WN1*CFL+CFQ*CFL*WN2)/(R*R)
      RETURN
      END
```

```
      COMPLEX FUNCTION ULN*16(LS,KSML)
      COMPLEX*16 UN,UNN,UNC,CS1,CS2,CMS,CM1
      COMPLEX EXCHI
      COMMON/SPAR/AKCI(6),TAU(6),SMLA(6),DMS(6),AK(4),AC(4),BJ(4),
     1CAPK,CAPC
      COMMON/ROTF/OM1,OM2,OMT
      COMMON/FREF/CMS,CM1,CS1,CS2
      COMMON/SWASH/SWGJ,SWEI,SWM,SWR
      COMMON/INTER/NSY,NBSEC,NFSEC,NB,NBP,MFLAP,MFEA,MCT,
     1MFLEX,MCON,MAER,MFUS,NBC,NFLAP,NFEA,NCT,NCON,NFFB,
     2NAS,NHC,NVI,NSP,MAXN,NES,MSC,NEGN,IPCT,NIT,MER,NORM,
     3IREM,NEX,NPS,NSCH,IG,IF,NPRL,NPRS,NPD,NSK,NCOLS,NCSB,
     4NFP1,MXSMI,MXT2P1,MXKQ,MXCPL,MXCSB,MXCPM,MXCPK,MXSMB,
     5NEBC,NESBC,MFASB,MXFAB,NFUS,NRBD,NRIFC,MXQ,NEIFC,
     6NEISC,NEITC,MXTKN,NFF,MINPN,MAXPN,IBF,MODE
      COMMON/NLEAD/NLEL,NLEL,MCTY
      UN =DCMPLX(0.D0,0.D0)
      UNN=DCMPLX(0.D0,0.D0)
      CFL=1.+(LS*LS-1)/(1+LS*LS*SWGJ/SWEI)
      CFLR=CFL/SWR
      CS1=CMS-CM1*KSML*OM1
      DO 5 JJ=1,NES
      UN=UN+(AK(JJ)+(CS1-CM1*LS*OM1)*AC(JJ))*EXCHI(0,LS,JJ)*
     1(1.-BJ(JJ)*CFLR)
    5 UNN=UNN+(AK(JJ)+(CS1-CM1*LS*OM1)*AC(JJ))
      IF(MSC.EQ.0) GO TO 6
      UNC=CAPK+(CS1-CM1*LS*OM1)*CAPC
      ULN=UN*UNC/(UNC+UNN)
    6 IF(MSC.EQ.0)ULN=UN
      RETURN
      END
```

```
      COMPLEX FUNCTION EXCHI (L,Q,J)
      INTEGER Q
      COMMON/RNAM1/CSA(4,24),SNA(4,24)
      LQ=L-Q
      ILQ=IABS(LQ)
      IF(LQ)16,15,17
   15 EXCHI=CMPLX(1.0,0.0)
      GO TO 18
   16 A=CSA(J,ILQ)
      B=SNA(J,ILQ)
      EXCHI=CMPLX(A,-B)
      GO TO 18
   17 A=CSA(J,ILQ)
      B=SNA(J,ILQ)
      EXCHI=CMPLX(A,B)
   18 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ZTEGI(I,J)
      INTEGER P,Q
      COMPLEX EXPON,EXPM1,EXPP1
      COMPLEX*16 CM1
      COMPLEX*16 B(648),SMLB(108),SMLC(108),SMLD(108)
      COMPLEX*16 CTB(54),FAB(54),FLB(54)
      COMPLEX*16 CTB1(9),CTB2(9),CTB3(9),FAB1(9),FAB3(9),FLB1(9),FLB2(9)
      COMPLEX*16 SMLE(108),FSB(54)
      COMPLEX*16 SMLF(108),SMLG(108)
      COMPLEX*16 FSB1(9),FSB2(9),FSB3(9),FAB4(9),FLB4(9),CTB4(9)
      COMPLEX*16 TKN(441)
      COMMON/BTS/B,SMLB,SMLC,SMLD,CTB,FAB,FLB,CTB1,CTB2,CTB3,FAB1,FAB3,
     1FLB1,FLB2,SMLE,FSB,FSB1,FSB2,FSB3,FAB4,FLB4,CTB4,SMLF,SMLG
      COMMON/TKN1/TKN
      COMMON/INTER/NSY,NBSEC,NFSEC,NB,NBP,MFLAP,MFEA,MCT,
     1MFLEX,MCON,MAER,MFUS,NBC,NFLAP,NFEA,NCT,NCON,NFFB,
     2NAS,NMC,NVI,NSP,MAXN,NES,MSC,NEGN,IPCT,NIT,MER,NORM,
     3IREM,NEX,NPS,NSCH,IG,IF,NPRL,NPRS,NPD,NSK,NCOLS,NCSB,
     4NFP1,MXSMI,MXT2P1,MXKQ,MXCPL,MXCSB,MXCPM,MXCPK,MXSMB,
     5NEBC,NESBC,MFASB,MXFAB,NFUS,NRBD,NRIFC,MXQ,NEIFC,
     6NEISC,NEITC,MXTKN,NFF,MINPN,MAXPN,IBF,MODE
      COMMON/NLEAD/MLEL,NLEL,MCTY
      CM1= DCMPLX(0.D0,1.D0)
      NMK= J-I
      KB= NEIFC+NEISC+MXT2P1+NFUS
      KBB= KB+NCOLS*NRIFC
      NMKP1=NMK+1
      NMKM1=NMK-1
      NPK=-NPS-I+NFP1
      NBM1=(NB-1)*NRBD
      LSMA=(J-1)*NESBC+(I-1)*MXCSB
      LLAR=(J-1)*NEBC+(I-1)*MXCPM
      IF(MFLEX.EQ.0) GO TO 10
      WRITE(6,900)
  900 FORMAT(/,9X,'GIMBALLED OR TEETERING ROTOR, MFLEX MUST EQUAL ZERO')
      GO TO 90
   10 IF(NBC.EQ.2)GO TO 13
      IF(NBP.EQ.0)GO TO 11
      IF(NBP.LE.2)GO TO 12
      GO TO 16
   11 IF(NB.GT.2)GO TO 16
   12 WRITE(6,901)
  901 FORMAT(/,9X,'GIMBALLED ROTOR MUST HAVE MORE THAN TWO BLADES')
      GO TO 90
   13 IF(NBP.EQ.0)GO TO 14
      IF(NBP.NE.2)GO TO 15
      GO TO 16
   14 IF(NB.EQ.2)GO TO 16
   15 WRITE(6,902)
  902 FORMAT(/,9X,'TEETERING ROTOR MUST HAVE TWO BLADES')
```

```
      GO TO 90
   16 NE=2
      NA=1
      IF(NBC.EQ.1)GO TO 17
      NE=1
      NA=0
   17 N1=-1
      N2=6
      DO 22 NN=1,NE
      IF(NN.EQ.1)GO TO 18
      N1=1
      N2=2
   18 DO 19 Q=1,NCOLS
      L3=LLAR+(Q-1)*12+3
      L10=L3+7
      KK=KB+(Q-1)*NRIFC+N2
   19 TKN(KK)=NA*B(L3)+N1*CM1*B(L10)
      L3=LSMA+3
      L10=L3+7
      KK=KBB+N2
      IF(MCT.EQ.0)GO TO 20
      TKN(KK)=-NA*SMLB(L3)-N1*CM1*SMLB(L10)
      KK=KK+NRIFC
   20 IF(MFEA.EQ.0)GO TO 21
      TKN(KK)=-NA*SMLC(L3)-N1*CM1*SMLC(L10)
      KK=KK+NRIFC
   21 IF(MFLAP.EQ.0) GO TO 23
      TKN(KK)=-NA*SMLD(L3)-N1*CM1*SMLD(L10)
      KK=KK+NRIFC
   23 IF(MLEL.EQ.0) GO TO 22
      TKN(KK)=-NA*SMLE(L3)-N1*CM1*SMLE(L10)
   22 CONTINUE
      IF(NBP.NE.0)GO TO 40
      IF(NBC.EQ.2)GO TO 28
      DO 24 Q=1,NRBD
      KT=KB+(Q-1)*NRIFC+2
      KK=KT+4
      K2=KT+NBM1
      K6=KK+NBM1
      TKN(K2)=-TKN(KT)
   24 TKN(K6)=-TKN(KK)
      DO 25 MS=2,NB
      EXPM1=EXPON(NMKM1,MS)
      EXPP1=EXPON(NMKP1,MS)
      MSHIFT=(MS-1)*NEITC+(MS-2)*NRBD
      DO 25 Q=1,NRBD
      KT=KB+(Q-1)*NRIFC+2
      KK=KT+4
      K2=KT+MSHIFT
      K6=KK+MSHIFT
```

```
      K22=K2+NRBD
      K66=K6+NRBD
      TKN(K2)=-TKN(KT)*EXPP1
      TKN(K6)=-TKN(KK)*EXPM1
      TKN(K22)=-TKN(K2)
   25 TKN(K66)=-TKN(K6)
      GO TO 100
   28 DO 30 Q=1,NCOLS
      L11=LLAR+(Q-1)*12+11
      KK=KB+(Q-1)*NRIFC+NRBD+6
   30 TKN(KK)=B(L11)
      L11=LSMA+11
      KK=KBB+NRBD+6
      IF(MCT.EQ.0)GO TO 31
      TKN(KK)=-SMLB(L11)
      KK=KK+NRIFC
   31 IF(MFEA.EQ.0)GO TO 32
      TKN(KK)=-SMLC(L11)
      KK=KK+NRIFC
   32 IF(MFLAP.EQ.0) GO TO 34
      TKN(KK)=-SMLD(L11)
      KK=KK+NRIFC
   34 IF(MLEL.EQ.0) GO TO 33
      TKN(KK)=-SMLE(L11)
   33 EXPM1=EXPON(NMKM1,2)
      DO 35 Q=1,NRBD
      KT=KB+(Q-1)*NRIFC+6
      KK=KT+NEITC
   35 TKN(KK)=-TKN(KT)*EXPM1
      DO 38 Q=1,NRBD
      KT=KB+(Q-1)*NRIFC+NRBD+6
      KK=KT+NEITC
   38 TKN(KK)=TKN(KT)*EXPM1
      GO TO 100
   40 N1=-1
      N2=6
      DO 50 NN=1,NE
      IF(NN.EQ.1)GO TO 42
      N1=1
      N2=2
   42 NPMK=NPK+N1
      INMK=IABS(NPMK)
      RINMK=1.0*INMK
      RFA=RINMK/NBP
      NFA=INMK/NBP
      DIF=ABS(RFA-1.0*NFA)
      IF(DIF.GT.(.05))GO TO 50
      DO 44 Q=1,NCOLS
      L4=LLAR+(Q-1)*12+4
      KK=KB+(Q-1)*NRIFC+N2
```

```
      L11=L4+7
   44 TKN(KK)=NA*B(L4)+N1*CM1*B(L11)
      L4=LSMA+4
      KK=KBB+N2
      L11=L4+7
      IF(MCT.EQ.0)GO TO 45
      TKN(KK)=-NA*SMLB(L4)-N1*CM1*SMLB(L11)
      KK=KK+NRIFC
   45 IF(MFEA.EQ.0)GO TO 46
      TKN(KK)=-NA*SMLC(L4)-N1*CM1*SMLC(L11)
      KK=KK+NRIFC
   46 IF(MFLAP.EQ.0) GO TO 47
      TKN(KK)=-NA*SMLD(L4)-N1*CM1*SMLD(L11)
      KK=KK+NRIFC
   47 IF(MLEL.EQ.0)GO TO 50
      TKN(KK)=-NA*SMLE(L4)-N1*CM1*SMLE(L11)
   50 CONTINUE
      GO TO 100
   90 STOP
  100 RETURN
      END
```

```
      SUBROUTINE POLAR
      COMPLEX QXJ,QYJ,QZJ,DTX,DTY,DTZ
      COMMON/QVTEM/QXJ,QYJ,QZJ
      DTX=QXJ*CMPLX(0.0,-1.0)
      DTY=QYJ*CMPLX(0.0,-1.0)
      DTZ=QZJ*CMPLX(0.0,-1.0)
      DXR=QXJ
      DYR=QYJ
      DZR=QZJ
      DXI=DTX
      DYI=DTY
      DZI=DTZ
      IF(DXR.NE.0.0) GO TO 2
      IF(DXI.NE.0.0) GO TO 2
      DXA=0.0
      GO TO 3
    2 DXA=ATAN2(DXI,DXR)
    3 IF(DYR.NE.0.0) GO TO 4
      IF(DYI.NE.0.0) GO TO 4
      DYA=0.0
      GO TO 5
    4 DYA=ATAN2(DYI,DYR)
    5 IF(DZR.NE.0.0) GO TO 6
      IF(DZI.NE.0.0) GO TO 6
      DZA=0.0
      GO TO 7
    6 DZA=ATAN2(DZI,DZR)
    7 CONTINUE
      DXR=SQRT(DXR*DXR+DXI*DXI)
      DYR=SQRT(DYR*DYR+DYI*DYI)
      DZR=SQRT(DZR*DZR+DZI*DZI)
      QXJ=CMPLX(DXR,DXA)
      QYJ=CMPLX(DYR,DYA)
      QZJ=CMPLX(DZR,DZA)
      RETURN
      END
```

```
*FORTRAN CALLABLE COMPLEX FUNCTION TO OBTAIN DOT PRODUCTS.
*ARGUMENT LIST IS (N,A,B), WHERE N IS THE DIMENSION OF THE VECTORS
*A AND B.  A IS PRESUMED SPARSE FOR MAXIMUM PROGRAM SPEED.
*INTERMEDIATE RESULTS ARE CARRIED IN DOUBLE PRECISION AND THE
*FUNCTION MAY BE DECLARED DOUBLE PRECISION COMPLEX, IF DESIRED.
*
          SPACE 2
#INCR     EQU   0
#COMPR    EQU   1
#INDEX    EQU   2
#N        EQU   2
#A        EQU   3
#B        EQU   4
#MAXR     EQU   4
          SPACE
#REAL     EQU   0
#IMAG     EQU   2
#ZERO     EQU   4
#TEMP     EQU   6
          SPACE 2
A         DSECT
AREAL     DS    D
AIMAG     DS    D
B         DSECT
BREAL     DS    D
BIMAG     DS    D
          EJECT
CDOT      CSECT
          SAVE  (2,#MAXR),,*
          USING CDOT,15
          LM    #N,#B,0(1)
          USING A,#A
          USING B,#B
          L     #COMPR,0(#N)
          BCTR  #COMPR,0
          SLA   #COMPR,4
          LA    #INCR,16
          SR    #INDEX,#INDEX
          SDR   #REAL,#REAL
          SDR   #IMAG,#IMAG
          SDR   #ZERO,#ZERO
          SPACE 2
LOOP      CD    #ZERO,A(#INDEX)
          BNE   CONTINUE
          BXLE  #INDEX,#INCR,LOOP
EXIT      RETURN (2,#MAXR)
          SPACE
CONTINUE  LD    #TEMP,AREAL(#INDEX)
          MD    #TEMP,BREAL(#INDEX)
          ADR   #REAL,#TEMP
```

```
          LD     #TEMP,AIMAG(#INDEX)
          MD     #TEMP,BIMAG(#INDEX)
          SDR    #REAL,#TEMP
          LD     #TEMP,AREAL(#INDEX)
          MD     #TEMP,BIMAG(#INDEX)
          ADR    #IMAG,#TEMP
          LD     #TEMP,AIMAG(#INDEX)
          MD     #TEMP,BREAL(#INDEX)
          ADR    #IMAG,#TEMP
          BXLE   #INDEX,#INCR,LOOP
          B      EXIT
          SPACE  2
          LTORG
          END                    (Start of RCDOT)
*FORTRAN CALLABLE COMPLEX FUNCTION TO OBTAIN DOT PRODUCTS.
*ARGUMENT LIST IS (N,A,B), WHERE N IS THE DIMENSION OF THE VECTORS
*A AND B.  A IS PRESUMED SPARSE FOR MAXIMUM PROGRAM SPEED.
*A IS A REAL VECTOR, WHILE B IS COMPLEX.
*INTERMEDIATE RESULTS ARE CARRIED IN DOUBLE PRECISION AND THE
*FUNCTION MAY BE DECLARED DOUBLE PRECISION COMPLEX, IF DESIRED.
*
          SPACE  2
#INCR     EQU    0
#COMPR    EQU    1
#INDEXA   EQU    2
#INDEXB   EQU    3
#N        EQU    3
#A        EQU    4
#B        EQU    5
#MAXR     EQU    5
          SPACE
#REAL     EQU    0
#IMAG     EQU    2
#ZERO     EQU    4
#TEMP     EQU    6
          SPACE  2
A         DSECT
B         DSECT
BREAL     DS     D
BIMAG     DS     D
          EJECT
RCDOT     CSECT
          SAVE   (2,#MAXR),,*
          USING  RCDOT,15
          LM     #N,#B,0(1)
          USING  A,#A
          USING  B,#B
          L      #COMPR,0(#N)
          BCTR   #COMPR,0
          SLA    #COMPR,3
```

```
          LA      #INCR,8
          SR      #INDEXA,#INDEXA
          SDR     #REAL,#REAL
          SDR     #IMAG,#IMAG
          SDR     #ZERO,#ZERO
          SPACE 2
LOOP      CD      #ZERO,A(#INDEXA)
          BNE     CONTINUE
          BXLE    #INDEXA,#INCR,LOOP
EXIT      RETURN (2,#MAXR)
          SPACE
CONTINUE  LA      #INDEXB,0(#INDEXA,#INDEXA)
          LD      #TEMP,A(#INDEXA)
          MD      #TEMP,BREAL(#INDEXB)
          ADR     #REAL,#TEMP
          LD      #TEMP,A(#INDEXA)
          MD      #TEMP,BIMAG(#INDEXB)
          ADR     #IMAG,#TEMP
          BXLE    #INDEXA,#INCR,LOOP
          B       EXIT
          SPACE 2
          LTORG
          END                    (Start of ROWSUM)
*FORTRAN CALLABLE SUBROUTINE TO PERFORM MATRIX ROW OPERATIONS.
*ARGUMENT LIST IS (N,NDIM,A,B,X).
*THE ROW OPERATION  A=A-X*B  IS PERFORMED, WHERE A, B, AND X ARE
*DOUBLE PRECISION COMPLEX.
*NDIM IS THE COLUMN DIMENSION OF THE MATRICES, AND N IS THE NUMBER OF
*ELEMENTS TO BE OPERATED ON IN THE ROWS.  THE INDEXING SCHEME IS
*THEREFORE  A(I)=A(I)-X*B(I),I=1,1+(N-1)*NDIM,NDIM
*
          SPACE 2
#INCR     EQU   0
#COMPR    EQU   1
#INDEX    EQU   2
#N        EQU   1
#NDIM     EQU   2
#A        EQU   3
#B        EQU   4
#X        EQU   5
#MAXR     EQU   5
          SPACE
#ATEMP    EQU   0
#BTEMP    EQU   2
#XREAL    EQU   4
#XIMAG    EQU   6
          SPACE 2
A         DSECT
AREAL     DS    D
AIMAG     DS    D
```

```
B         DSECT
BREAL     DS    D
BIMAG     DS    D
          EJECT
ROWSUM    CSECT
          USING ROWSUM,15
          SAVE  (14,#MAXR),,*
          LM    #N,#X,0(1)
          USING A,#A
          USING B,#B
          L     #NDIM,0(#NDIM)
          SLA   #NDIM,4
          L     #COMPR,0(#N)
          BCTR  #COMPR,0
          MR    #COMPR-1,#NDIM
          LR    #INCR,#NDIM
          SR    #INDEX,#INDEX
          LD    #XREAL,0(#X)
          LD    #XIMAG,8(#X)
          SPACE 2
LOOP      LD    #ATEMP,BIMAG(#INDEX)
          MDR   #ATEMP,#XIMAG
          AD    #ATEMP,AREAL(#INDEX)
          LD    #BTEMP,BREAL(#INDEX)
          MDR   #BTEMP,#XREAL
          SDR   #ATEMP,#BTEMP
          STD   #ATEMP,AREAL(#INDEX)
          LD    #ATEMP,AIMAG(#INDEX)
          LD    #BTEMP,BREAL(#INDEX)
          MDR   #BTEMP,#XIMAG
          SDR   #ATEMP,#BTEMP
          LD    #BTEMP,BIMAG(#INDEX)
          MDR   #BTEMP,#XREAL
          SDR   #ATEMP,#BTEMP
          STD   #ATEMP,AIMAG(#INDEX)
          BXLE  #INDEX,#INCR,LOOP
          SPACE
          RETURN (2,#MAXR),T
          LTORG
          END
 OVERLAY ALPHA                    <-------  (Start of Overlay Structure)
 INSERT ARI
 INSERT ARR
 INSERT COEFFS
 INSERT BTS
 INSERT SS1
 INSERT TKN1
 OVERLAY BETA
 INSERT BAERO
 INSERT FAERO
```

```
INSERT AERO
INSERT ACOEFF
INSERT SETUP
INSERT SECPAR
INSERT LOADIN
INSERT TABLU
OVERLAY BETA
INSERT ELAST
INSERT RIGID
INSERT STIFF
INSERT BEND
INSERT MLPC2
INSERT MLCC2
INSERT CDOT
INSERT RCDOT
OVERLAY GAMMA
INSERT BARRAY
INSERT BMASS
INSERT BLARO
OVERLAY GAMMA
INSERT SARRAY
INSERT EMASS
INSERT EUARO
OVERLAY BETA
INSERT EPSOLN
INSERT SUBMA
INSERT SUBMB
INSERT SUBME
INSERT SUBMG
INSERT ZTEGI
INSERT EXPON
INSERT TKNS
INSERT SWA
INSERT SWB
INSERT ZIN
INSERT XNLO
INSERT ULN
INSERT EXCHI
OVERLAY ALPHA
INSERT SOLVE
INSERT DCMAT
INSERT SWAPS
INSERT ROWSUM
INSERT ERRSET
ENTRY MAIN
```

REFERENCES

1. Sutton, Lawrence R. and Gangwani, Santu T., THE DEVELOPMENT AND APPLICATION OF AN ANALYSIS FOR THE DETERMINATION OF COUPLED TAIL ROTOR/HELICOPTER AIR RESONANCE BEHAVIOR, USAAMRDL-TR-75-35, Eustis Directorate, US Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia, August, 1975, AD AO14989.

2. Sutton, Lawrence R. and White, Richard P. Jr., THE DEVELOPMENT AND APPLICATION OF AN ANALYSIS FOR THE STABILITY BEHAVIOR OF BEARINGLESS MAIN ROTOR SYSTEMS, Letter Report SRL 14-77-3, RASA Division, Systems Research Laboratories, Inc., 1055 J. Clyde Morris Blvd., Newport News, Virginia.

3. Davis, John M., ROTORCRAFT FLIGHT SIMULATION WITH AEROELASTIC ROTOR AND IMPROVED AERODYNAMIC REPRESENTATION, Volume II - User's Manual, Bell Helicopter Company; USAAMRDL-TR-74-10B, Eustis Directorate, US Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia, June 1974, AD 782756.